

UNITED STATES PATENT APPLICATION

for

**CHECK BOXES FOR IDENTIFYING  
AND PROCESSING STORED DOCUMENTS**

Inventor:

John Barrus

prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP  
12400 Wilshire Boulevard  
Los Angeles, CA 90025-1026  
(408) 720-8300

File No.: 074451.P161

**EXPRESS MAIL CERTIFICATE OF MAILING**

"Express Mail" mailing label number: EV 409 357 436 US

Date of Deposit: March 31, 2004

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service  
"Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has  
been addressed to the [INSERT NEW PTO ADDRESS]

Angela M. Quinn

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

(Date signed)

March 31, 2004

CHECK BOXES FOR IDENTIFYING AND PROCESSING STORED DOCUMENTS

**BACKGROUND OF THE INVENTION**

[0001] This application is a continuation-in-part of U.S. Patent Application No. \_\_\_\_\_, entitled "ACTION STICKERS FOR IDENTIFYING AND PROCESSING STORED DOCUMENTS," filed September 16, 2003, which is a continuation-in-part of U.S. Patent Application No. 10/404,916 entitled "METHOD AND APPARATUS FOR COMPOSING MULTIMEDIA DOCUMENTS," filed March 31, 2003.

[0002] This application is related to the following commonly owned and co-pending U.S. patent applications:

- U.S. Patent Application No. 10/404, 927 entitled "MULTIMEDIA DOCUMENT SHARING METHOD AND APPARATUS," filed March 31, 2003;
- U.S. Patent Application No. 09/521, 252 entitled "METHOD AND SYSTEM FOR INFORMATION MANAGEMENT TO FACILITATE THE EXCHANGE OF IDEAS DURING A COLLABORATIVE EFFORT," filed March 8, 2000;
- U.S. Patent Application No. 10/001,895 entitled "PAPER-BASED INTERFACE FOR MULTIMEDIA INFORMATION," filed November 19, 2001;
- U.S. Patent Application No. 10/081,129 entitled "MULTIMEDIA VISUALIZATION & INTEGRATION ENVIRONMENT," filed February 21, 2002;

- U.S. Patent Application No. 10/085,569 entitled “A DOCUMENT DISTRIBUTION AND STORAGE SYSTEM,” file February 26, 2002;
- U.S. Patent Application No. 10/174,522 entitled “TELEVISION-BASED VISUALIZATION AND NAVIGATION INTERFACE,” filed June 17, 2002;
- U.S. Patent Application No. 10/175,540 entitled “DEVICE FOR GENERATING A MULTIMEDIA PAPER DOCUMENT,” filed June 18, 2002;
- U.S. Patent Application No. 10/307,235 entitled “MULTIMODAL ACCESS OF MEETING RECORDINGS,” filed November 29, 2002; and
- U.S. Patent Application No. \_\_\_\_\_ entitled “PHYSICAL KEY FOR ACCESSING A SECURELY STORED DIGITAL DOCUMENT,” filed August 11, 2003.

### **FIELD OF THE INVENTION**

[0003] This invention relates generally to document management, and more specifically to techniques of identifying documents in a digitally stored collection and specifying actions to execute on the documents.

### **BACKGROUND OF THE INVENTION**

[0004] Despite the ideal of a paperless environment that the popularization of computers had promised, paper continues to dominate the office landscape. Ironically, the computer itself has been a major contributing source of paper proliferation. The

computer simplifies the task of document composition, and thus has enabled even greater numbers of publishers. Oftentimes, many copies of a document must be made so that the document can be shared among colleagues, thus generating even more paper.

[0005] Despite advances in technology, practical substitutes for paper remain to be developed. Computer displays, PDAs (personal digital assistants), wireless devices, and the like all have their various advantages, but they lack the simplicity, reliability, portability, relative permanence, universality, and familiarity of paper. In many situations, paper remains the simplest and most effective way to store and distribute information.

[0006] The conveniences and advantages that paper offers signal that its complete replacement is not likely to occur soon, if ever. Perhaps then, the role of the computer is not to achieve a paperless society. Instead, the role of the computer may be as a tool to move effortlessly between paper and electronic representations and maintain connections between the paper and the electronic media with which it was created.

[0007] Related, commonly owned, above-referenced patent application serial nos. 10/404,916 and 10/404, 927 describe techniques for organizing multimedia documents into one or more collections. A collection coversheet, or document index, representative of the collection can be printed on a suitable medium, such as paper. This coversheet can provide access to the collection by using a multi-function peripheral (MFP). In this way, individuals can share the multimedia documents in the collection by distributing copies of the coversheet to recipients.

[0008] Most prior methods to interact with digitally stored documents require the user to enter commands by typing or pressing buttons on hardware or selecting options

from displayed menus on the MFP or on a computer. These systems require the user to interact with the hardware and/or navigate menu options and other user interface features on a display device. Some existing paper-based systems require specialized coversheets to provide processing instructions. For example, a coversheet may be used at the beginning of the print job to specify the number of copies, the size of the paper, etc. These systems require a supply of these coversheets to be kept on hand, and usually require the user to take the time to customize the sheet by filling in the details of the job.

[0009] The FlowPort system of Xerox provides three different types of paper interfaces. A FlowPort Cover Sheet provides instructions to a scanning system, a Document Token stands in place of a single multi-page document, and a Document Catalog having a linear list of file names can be used to select more than one document using a single sheet of paper. The FlowPort Cover Sheet is a list of destinations and categories. The Cover Sheet can be used to indicate how to route the documents that follow. The document might be e-mailed, faxed, printed, or categorized. Each of the destinations on the Cover Sheet has the appropriate fax number, e-mail address, or printer address associated with it in advance. Cover Sheets are placed on the top of documents, Document Tokens, or Document Catalogs before scanning. The Cover Sheet must be created at the computer and not generated at a multi-function peripheral (MFP). The FlowPort Document Token is a document token representing a single multi-page document. A thumbnail of the first page of the document is displayed as well as the document's machine readable index into the local Xerox DocuShare database. The Token page can be used as a stand-in for a document that already exists in the DocuShare database. The FlowPort Document Catalog is a page containing a linear list of names of

documents stored in a DocuShare repository. The check box next to the document name allows the user to select some of the documents in the catalog to be routed using a Cover Sheet. Each Document Catalog has a machine readable index to the collection of documents.

**[0010]** Each FlowPort operation requires at least two sheets of paper, including one generated by a desktop computer. That is, with FlowPort, the Cover Sheet is generated at a computer. The user starts by specifying destinations and creating categories and prints out the task-specific Cover Sheet. This Cover Sheet can be placed on top of a Document Token or a Document Catalog. All the documents represented by the token or selected by check mark on the catalog are routed as indicated on the Cover Sheet.

**[0011]** Xerox Research Centre Europe Project KnowledgePump allows researchers to exchange, discuss, and recommend documents in web pages. KnowledgePump permits the addition of comments and the classification of documents using Cover Sheets. Each Cover Sheet includes a thumbnail of the first page of the document, an index to its electronic counterpart, and space for handwritten notes. Check boxes are provided for ranking and classification of the document. For instance, a user can mark the "very interesting" box if the article is found to be useful and interesting. When the Cover Sheet is scanned, if check boxes are marked, the database entry for that document is updated to reflect the selections indicated by the user.

**[0012]** What is needed is a system and method for providing instructions for processing documents without requiring users to interact with a user interface or hardware device. What is further needed is a system and method that avoids the

limitations of prior art schemes for providing instructions for processing stored documents.

## **SUMMARY**

**[0013]** A method and apparatus for identifying and processing stored documents.

In one embodiment, the method comprises receiving an image of an overview of a collection and a machine readable pointer identifying the collection, identifying at least one action set forth in the image, identifying at least one document, and performing the at least one action on the at least one document.



## **BRIEF DESCRIPTION OF THE DRAWINGS**

**[0014]** The accompanying drawing illustrates several embodiments of the invention and, together with the description, serves to explain the principles of the invention.

**[0015]** Fig. 1 is a block diagram of one embodiment of an architecture of a system for reading check boxes and performing actions on stored documents in response to marked check boxes.

**[0016]** Fig. 2 is a flow diagram of one embodiment of a process for reading check boxes and performing actions on stored documents responsive to check boxes being selected.

**[0017]** Fig. 3 is an example depicting check boxes on a coversheet of a collection.

## **DETAILED DESCRIPTION OF THE EMBODIMENTS**

**[0018]** A method and apparatus for using collection coversheets with check boxes is disclosed. The collection coversheet may use thumbnails to represent documents and optionally may have titles, which are unrelated to their filenames. In one embodiment, a user selects one or more check boxes on a collection coversheet to identify, by location on the coversheet, target documents within a previously stored collection of documents. In one embodiment, each selection can be accessible using un-guessable, unique identifiers anywhere on the Internet. The check boxes also specify actions to be performed on the target documents. The coversheet is scanned and the check boxes are located and read to determine which have been marked. The specified actions are then performed on the target documents. If the specified actions change the organization or architecture of the document collection, an updated version of the collection may be generated and stored, and a new coversheet may be printed. Thus, the actions and document selection may be completed using a single sheet. The collections can contain other collections. That is, the document in a sub-collection can be printed or e-mailed using the coversheet of the containing collection. Thus, the selection of the check box may select a document contained in a sub-collection.

**[0019]** Note that file names are not used on the coversheet. Thus, the entire process can be performed at an MFP. That is, the collection can be created and printed out with check boxes, and then an action can be selected using a check box and put through the MFP.

**[0020]** Check boxes for identifying and processing stored documents are described. In the following description, numerous details are set forth, such as distances

between sizes of check boxes, location of check boxes, etc. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form, rather than in detail, in order to avoid obscuring the present invention.

**[0021]** Reference in the specification to “one embodiment” or “an embodiment” means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of the phrase “in one embodiment” in various places in the specification are not necessarily all referring to the same embodiment.

**[0022]** Some portions of the detailed descriptions that follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

**[0023]** It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the

following discussion, it is appreciated that throughout the description, discussions utilizing terms such as “processing” or “computing” or “calculating” or “determining” or “displaying” or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

**[0024]** The present invention also relates to apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus.

**[0025]** The algorithms and modules presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description below. In addition, the present invention is not described with reference to any particular

programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein.

**[0026]** A machine-readable medium includes any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine-readable medium includes read only memory (“ROM”); random access memory (“RAM”); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

**[0027]** In this application, the following terms are used:

**[0028]** “Document” refers to any collection of information capable of being stored electronically, including but not limited to text, word processing and spreadsheet files, e-mail messages, voice and audio recordings, images, archives of documents, and video recordings.

**[0029]** “Identifier sheet” refers to a piece of paper or other readable media item that identifies a stored document or collection of documents. As described in above-reference related patent applications, the identifier sheet may be a collection coversheet or may take on any other form. In one embodiment, the identifier sheet includes a document identifier and/or collection identifier that may be computer-readable, human-readable, or any combination thereof. Identifier sheets are also referred to herein as “document indexes.”

**[0030]** One type of identifier sheet is a “collection coversheet.” A collection coversheet identifies a collection and also includes representations of documents within the collection. In one embodiment, a collection coversheet includes:

- A collection identifier in machine-readable form (such as a barcode) and/or human-readable form (such as a Uniform Resource Locator (URL) or other text string). The collection identifier provides information describing a location of the collection, such as a directory or folder containing documents in the collection.
- A collection overview, which represents documents in the collection by thumbnails. Thumbnails are associated with positions in the overview. For instance, the thumbnail for document A might be in the upper left corner of the collection overview, and the thumbnail for document B might be in the lower right corner.

**[0031]** Further description of collection coversheets, collection identifiers, and collection overviews can be found in related patent applications referenced above and are discussed in more detail below.

**[0032]** For illustrative purposes, the following description sets forth the invention in terms of check boxes and other indication or selection areas on collection coversheets. However, one skilled in the art will recognize that the invention can also be implemented using check boxes on other types of identifier sheets, document indexes, or media items that identify stored documents, and that such implementations would not depart from the essential characteristics of the present invention.

**[0033]** Referring now to Fig. 3, there is shown an example of check boxes appearing on a collection coversheet 101. In one embodiment, collection coversheet 101 is a piece of paper that includes machine-readable collection identifier 102 and collection overview area 501 containing thumbnail representations 503A-H of digital documents. Also included is action indication area 510 where actions may be specified and

annotation area 502 where notes may be written. The particular layout and components shown in Fig. 3 are merely exemplary.

**[0034]** In the example of Fig. 3, check boxes 501<sub>1</sub>, 503A<sub>1</sub>-503H<sub>3</sub>, and 515 are located on coversheet 101. Check boxes 503A<sub>1</sub>-503H<sub>3</sub> are printed on thumbnail representations 503A-H, respectively, which refers to one of the documents in the collection associated with coversheet 101. Thumbnail representation 503H includes three check boxes 503H<sub>1</sub>-503H<sub>3</sub>. Thumbnail 503H represents a sub-collection where each of check boxes 503H<sub>1</sub>-503H<sub>3</sub> corresponds to one of the documents (or collection of documents) in the sub-collection of 503H.

**[0035]** Check box 501<sub>1</sub> appears on collection overview area 501, and check boxes 515 appear on action indication area. In general, check boxes 503A<sub>1</sub>-503H<sub>3</sub> may be located anywhere in overview area 501 in close proximity to one of the thumbnail representations to ensure a user desiring an action to be performed to a particular document is able to determine the correct check box to mark their selection. Similarly, each of check boxes 515 is located in close proximity to a printed word specifying an action to allow a user to select one or more actions to be performed on selected documents.

**[0036]** The system of the present invention is capable of recognizing selected, or marked, check boxes regardless of the marks made in them. The user may mark the check box by filling in the check box or putting another mark (e.g., a check mark) in the box. In one embodiment, if any of the checkboxes has a dark pixel, it is considered marked. In consideration of the fact that sometimes small bits of dust or noise in the scanning mechanism might form a dark pixel, it is desirable to remove individual dark

pixels or small dots from the checkbox image before deciding if it has been marked.

Removing noise from images is well understood in the art and is typically done using morphological operations. For a description of morphological operations, see Dougherty, Edward R; Jaakko Astola, "An Introduction to Nonlinear Image Processing" (Vol TT16) Tutorial texts in optical engineering, O'Shea, Donald ed., SPIE Optical Engineering Press, Bellingham, WA, 1994.

**[0037]** Check boxes 515 as shown in Fig. 3 are square shaped and are next to, or in close proximity to, a word indicating the desired action (e.g., "Print"). Other formats and shapes (e.g., circles, ovals, etc.) are possible and will be recognized by one skilled in the art in light of this description. For example, another form of indication may be circular or elliptical to provide a user an area to mark to specify an action.

**[0038]** In one embodiment, such an arrangement would signal to the MFP of the present invention that the requested action should be performed on those documents that correspond to thumbnails located between the check boxes.

**[0039]** One of check boxes 515 indicates that the user wishes to perform a grouping action and any of check boxes 503A-H marked would identify particular documents that the user wishes to group together as a sub-collection within the original collection represented by coversheet 101. The MFP interprets the marked check boxes and performs the grouping operation as requested. In one embodiment, grouping and sub-collection organization is implemented as described in related cross-referenced patent applications. The grouping operation consists of creating a new collection, moving the two documents or media into the new collection by adding them to the new



collection and deleting them from the old collection. The new collection is then added to the old collection in approximately the same location as the original files.

**[0040]** Collection check box 503H<sub>1</sub> may be marked to select the collection. This is different than selecting all the documents in the collection. If the collection check box 501H<sub>1</sub> is checked and the action “print” is selected by the user, the coversheet 101 is printed. If the collection check box 503H<sub>1</sub> is selected and the operation “delete” is selected, the subcollection will be deleted from the main collection instead of deleting the individual documents contained in the subcollection.

**[0041]** It may be desirable for the user to print out not just a single collection coversheet but also all of the documents contained in that collection. If that collection contains a subcollection, it may also be desirable to print out the documents contained in the subcollection also. In one embodiment, if the user has indicated that an action should be performed on an entire collection and when it is possible to perform that action on other collections contained within that collection, it would be beneficial to give the user the opportunity to perform that action on the entire hierarchy. Of course, if the collection represents the root collection of a deep hierarchy of collections, the user may choose to limit the depth of the action so that it terminates before it reaches all of the documents.

**[0042]** For instance, if the user would like to print a collection, the documents in that collection and all of the documents contained in that collection’s subcollections, he could select the print action and choose a depth of 2. At depth 1, all of the documents and coversheets of contained collections are printed. At depth 2, all of the documents inside of the contained collections are also printed.

**[0043]** The maximum depth of an action could be indicated using the control panel of the MFP or “Depth” checkboxes could be provided in the paper user interface on the collection coversheet.

**[0044]** In one embodiment, locating marked check boxes is performed using morphological operations well known to those skilled in the art. More specifically, a program performing morphological operations takes an input image and a “kernel” that resembles the object that is being sought, namely a check box of a given size in this case and compares the kernel with the input image at every pixel. Every time the kernel at a given position in the image exactly matches the input image, it leaves a dark pixel in the output image. In other words, the output image has a dark pixel in every place a check box appears in the input image. Processing logic can search for pixels in the outcome image after comparison and produce a list of image coordinates where there are marked check boxes.

**[0045]** Since a map exists for the collection, processing logic can find the collection overview and will look in the corners of the targets or objects. Similarly, it can keep track of where the boxes are printed and look precisely for the boxes later.

**[0046]** Referring now to Fig. 1, there is shown a block diagram depicting one embodiment of a functional architecture of a system for reading marked check boxes and performing actions on stored documents responsive to the marked check boxes.

Referring also to Fig. 2, there is shown a flow diagram of one embodiment of a process for reading marked check boxes and performing actions on stored documents responsive to the marked check boxes. The process may be performed, for example, by the system depicted in Fig. 1, or by other functional components and systems. The order of the

operations in the described embodiment is merely exemplary. One skilled in the art will recognize that the operations can be performed in an order other than what is depicted.

**[0047]** The use of check boxes is described herein in the context of a multifunction peripheral (MFP) 100 including scanner 104, a machine-readable code locator and reader 106, a marked check box locator 107, a document identifier and processor 113, and printer 115. Marked check box locator 107 may also include functionality for locating collection overview area 501 within collection coversheet 101; alternatively, such functionality may be provided in a separate component (not shown).

**[0048]** MFP 100 may also contain other components, some of which may not be required for the operation of this invention. MFP 100 may contain a network interface card (not shown), which can receive processing requests from the external network, a fax interface, media capture devices, a media capture port, and the like.

**[0049]** Control interface 117 provides a mechanism by which the user can initiate, configure, monitor, and/or terminate MFP 100 operations, for example, to make copies, scan documents, and print faxes. In one embodiment, interface 117 includes a keypad, display, touchscreen, or any combination thereof.

**[0050]** The components shown in MFP 100 are functional components that may be implemented using any combination of hardware elements, software, or the like. For example, the functionality of reader 106 and locator 107 may be implemented within a single hardware component and/or software module, or they may be broken out into separate functional components. Accordingly, the architecture shown in Fig. 1 is intended to illustrate the overall functionality of the invention according to one

embodiment, and is not intended to limit the scope of the claimed invention to any particular set of components.

**[0051]** In one embodiment, MFP 100 can access other forms of media through electronic data input peripherals (not shown) including, for example, magnetic media readers for magnetic media such as floppy disks, magnetic tape, fixed hard disks, removable hard disks, memory cards, and the like. Peripherals may also include optical media readers (not shown) for optical storage media such as CDs, DVDs, magneto-optical disks, and the like. In addition, in one embodiment MFP 100 is communicatively coupled to storage device 109, which may be a hard drive or other device capable of storing collections of digital documents, for example in database form. Storage device 109 may be at the same location as MFP 100, or it may be remotely located, connected for example via a network.

**[0052]** As described above in connection with Fig. 3A, collection coversheet 101 includes machine-readable collection identifier 102 and collection overview area 501 containing thumbnail representations 503A-H of digital documents. Alternatively, collection coversheet 101 may have an embedded RFID tag containing collection identifier 102. Check boxes are included on coversheet 101 and may be marked to point to one of thumbnails 503A-H, thus identifying a particular document as the target for an action specified by marking one or more of the check boxes in action selection area.

**[0053]** MFP 100 receives 201 an image 105 of coversheet 101, for example by scanning coversheet 101 using scanner 104 according to techniques that are well known in the art. Alternatively, MFP 100 may use other input mechanisms known to persons of ordinary skill in the art to receive the image of coversheet 101 (processing block 201).

For example, MFP 100 may receive the image via e-mail, fax, retrieval from previously stored coversheet 101 images, or the like.

**[0054]** MFP 100 then locates 202 collection identifier 102 within image 105 of coversheet 101, and reads the identifier 102 (processing block 203). In one embodiment, processing blocks 202 and 203 are performed by passing image 105 or the physical page in the case of RFID to code locator and reader 106, which locates and reads collection identifier 102. Collection identifier 102 identifies the storage location of documents in the collection. In one embodiment, identifier 102 is a URL or the like that identifies documents by location and filename. For example, identifier 102 may identify documents within storage device 109. In one embodiment, identifier 102 also identifies a map that associates documents with particular regions within collection overview 501.

**[0055]** Code locator and reader 106 passes the read collection identifier 102 to document identifier and processor 113 as described in more detail below.

**[0056]** MFP 100 locates 204 collection overview 501 within image 105 of coversheet 101, for example by determining the overall size and shape of overview 501. In one embodiment, overview 501 is provided at a standard location within coversheet 101, or is color-coded or otherwise marked, so as to facilitate easier identification of overview 501. Alternatively, overview 501 can be at an arbitrary location and have arbitrary characteristics. In one embodiment, marked check box locator 1076 component of MFP 100 performs processing block 204; in another embodiment, another component (not shown) of MFP 100 performs this operation.

**[0057]** MFP 100 locates 205 check box(s) that have been marked on collection overview 501. In one embodiment, marked check box locator 107 component of MFP

100 performs processing block 205. Processing block 205 may be performed in response to the user specifying, via control interface 117, that one or more marked check boxes are present. Alternatively, locator 107 may be configured to automatically attempt to locate marked check boxes whenever a coversheet 101 has been scanned by scanner 104.

**[0058]** In one embodiment, marked check boxes are recognized by check box locator 107. Alternative methods for locating objects in an image are known in the art or have been described in related co-pending applications.

**[0059]** Based on which of the check boxes have been marked in the action indication area of coversheet 101, check box locator 107 identifies 206 the desired action(s). In one embodiment, action sticker locator and reader 107 passes the action request 112 to document identifier and processor 113.

**[0060]** MFP 101 also determines, based on which of the check boxes 503A<sub>1</sub>-503H<sub>3</sub> in overview 501 are marked, the desired target document(s) for the action. In one embodiment, check box locator 107 determines the location of each marked check box 503A<sub>1</sub>-503H<sub>3</sub>, and document identifier and processor 113 determines a target document by comparing the location of marked check box with known information about thumbnail 503 locations in overview 501.

**[0061]** In one embodiment, storage device 109 includes a map 110 corresponding to each collection; the map provides coordinates for thumbnails 503 within overview 501. Thus, two-dimensional coordinates within overview 501 identify (or map to) documents, based on the locations of thumbnails 503 for those documents. In one embodiment, the map is implemented as a list of rectangles, one representing the entire collection overview 501, and other rectangles representing positions of document thumbnails 503

within the overview 501. Map 110 may be stored as a separate file, such as a Scalable Vector Graphics (SVG) file containing a description of collection overview 501 with identifiers that associate regions within the overview 501 with documents in the collection. Alternatively, map 110 may be stored as part of collection information 301.

**[0062]** Document identifier and processor 113 uses collection identifier 102 (obtained from code locator and reader 106) to retrieve, from storage 109, map 110 indicating the correspondence of coordinates within collection overview 501 to collection documents. Based on the map and based on marked check box location information 111, document identifier and processor 113 determines a target document. If a marked check box is within a rectangle representing a document thumbnail 503, the corresponding document is deemed to be the target of the action. Alternatively, in such a situation where ambiguity exists as to whether a document is the target document, MFP 101 can do any of the following: prompt the user, via control interface 117, to specify whether the document is intended to be the target.

**[0063]** In one embodiment, processing blocks 205 and 206 are performed using known techniques of optical feature recognition.

**[0064]** If more than one check box 103 is found marked in action indication, document identifier and processor 113 sorts the actions in an appropriate order (processing block 207). For example, if marked check boxes 103 in the action indication area indicate that one or more documents should be both printed and deleted, the print action should take place before the delete action. In one embodiment, the default sort order is as follows: print, e-mail, fax, group, ungroup, delete. Alternatively, MFP 100 may alert the user to the presence of multiple actions on a document and request

clarification (via control interface 117, for example) as to the intended order to carry out the actions.

**[0065]** If a specified action involves transmitting the document, for example, by e-mail or fax, MFP 100 locates the correct routing information (such as an e-mail address or a fax number) indicating the desired destination for the document. Routing information can be included on or written in the action indication area 515, or some other predetermined area on coversheet 101, such as written in annotation area 502 of coversheet 101, so that it can be extracted via optical character recognition (OCR). For example, if a single e-mail address is written in action indication area 515 or on the line next to the e-mail action, all documents to be e-mailed can be sent to that e-mail address. Alternatively, MFP 100 can prompt the user to enter routing information via control interface 117. Alternatively, the routing information could be written on a second sheet of paper to be scanned or in a second image received by MFP 100. In any of these embodiments, the operation of determining routing information can be performed by marked check box locator 107, or by document identifier and processor 113, or by another component of MFP 100.

**[0066]** Once actions and target document(s) have been determined, document identifier and processor 113 uses collection identifier 102 (obtained from code locator and reader 106) to retrieve, from storage 109 (processing block 208), the target document(s) 114 and performs the specified action(s) in the determined order (processing block 209). For some actions (such as delete), retrieval of the document(s) 114 is not required, and processing block 208 is not performed. In one embodiment, document



identifier and processor 113 first retrieves collection information 301 which includes or points to target document(s) 114, and then obtains target document(s) 114 accordingly.

**[0067]** Some examples of check boxes and their corresponding actions include:

**[0068]** Print check box: Document identifier and processor 113 sends target document(s) 114 to printer 115. Printer 115 outputs printed document 116.

**[0069]** E-mail or fax check box: Document identifier and processor 113 sends the documents to an e-mail or fax module (not shown) of MFP 100 to be transmitted accordingly.

**[0070]** Group check box: Document identifier and processor 113 creates a new sub-collection including the target documents, deletes the target documents from the original collections, and adds the new sub-collection to the original collection. In one embodiment, all of the documents identified with marked check boxes are placed into the same new sub-collection.

**[0071]** Ungroup check box (on an existing sub-collection): Documents within the sub-collection are retrieved and placed in the overall collection corresponding to coversheet 101.

**[0072]** Delete check box: Document identifier and processor 113 deletes the specified document(s) or sub-collection(s). In one embodiment, a confirmation dialog box is presented on control interface 117 before the delete operation is performed.

**[0073]** Play check box: Document identifier and processor 113 sends target document(s) 114 (such as audio and/or video files) to an output device to be played, such as audio I/O device 120.

**[0074]** Document identifier and processor 113 determines 210 whether any of the performed actions cause changes to collection map 110 and overview 501. If so, document identifier and processor 113 modifies 211 collection map 110 and overview 501 accordingly to indicate locations of thumbnails 503A-H corresponding to new documents and sub-collections and to delete one or more thumbnails 503A-H for documents and sub-collections that have been removed. The updated collection info 301, map 110, and/or overview 501 are stored in storage device 109. Optionally, the updated collection information 301 and map 110 are sent to coversheet generator 302 for generation of an updated coversheet 101A including a new overview 501, as described below.

**[0075]** When documents are moved from one collection to another, a default layout can be used for the arrangement of thumbnails 503A-H. Alternatively, the user may be given an opportunity to indicate a layout. Such techniques are described in related cross-referenced patent applications.

**[0076]** Printer 115 may optionally (or automatically) print 212 a new collection coversheet 306 representing the collection, particularly if collection organization has been modified, or if check boxes have been marked.

**[0077]** For example, printing of a document in the collection can be requested by marking a print check box on a coversheet 101. Machine-readable code locator and reader 106 reads the collection identifier 102. Check box locator 107 locates and reads the marked print check box, passing marked check box location information 111 and a print action request 112 to document identifier and processor 113. Document identifier and processor 113 identifies the target document based on the marked check box location

information 111 and on map 110 retrieved from storage 109. Document identifier and processor 113 retrieves document 114 from storage and passes it to printer 115. Printer 115 outputs printed document 116.

**[0078]** In one embodiment, when collection organization is modified (such as by changing hierarchy, layout, or access levels), a new version of the collection is created. Thus, rather than overwriting the collection with new information, an updated version of the collection is generated and stored in a new location within storage 109, and a new collection identifier 102 is generated that points to the new location. A new coversheet 101A is printed with the new collection identifier 102. In this manner, previous versions of collections are preserved.

**[0079]** For example, when a document is deleted, a new collection is created which is exactly like the original collection except that it omits the deleted document. Map 110 and overview 501 are altered to reflect that the document has been deleted. The new collection can be a new version of the original collection. Such versioning techniques are described in detail in related cross-referenced applications.

**[0080]** In one embodiment, MFP 100 includes coversheet generator 302, either as a separate functional module or as a component of document identifier and processor 113 or some other component. Coversheet generator 302 is therefore an optional component that need not be included, and indeed is absent in some embodiments. When included, coversheet generator 302 performs processing block 211 to receive updated collection info 301A from document identifier and processor 113, modify collection map 110, and generate an updated coversheet 101A to be sent to printer 115 to be output as printed coversheet 306.

### Coversheets

**[0081]** A collection coversheet is a paper that represents a collection and, in one embodiment, comprises a header, a string of text printed in a machine-readable format, a collection overview image, optionally, an area in which notes may be written, and optionally a human-readable version of the text encoded in the machine-readable code.

**[0082]** The header contains printed information about the collection. This information may include the author of the collection, a list of zero, one or more people who will be notified if the collection is modified, time and date information about when the collection was last modified or when this coversheet was printed out, and an optional collection topic or subject.

**[0083]** In one embodiment, the machine-readable code contains an encoded version of a unique pointer to the collection on the collection server. In one embodiment, this same pointer when presented in the human-readable form is similar to a uniform resource locator or URL used in the World Wide Web and is referred to herein as a collection identifier, distributed resource identifier, or DRI. In one embodiment, a collection server uses these DRIs as unique collection pointers. In one embodiment, DRIs are globally unique, difficult to guess, and can provide access to collections from anywhere on the Internet.

**[0084]** Within this specification, the terms “collection identifier,” “distributed resource identifier,” and “DRI” will be used interchangeably and should be understood to mean the same thing – a unique identifier that points to a collection of media and documents stored on a collection server. Also, the identifier might be written in human-

readable form or machine-readable form. Both printed forms represent the same identifier and point to the same collection even though they look unlike each other.

**[0085]** In one embodiment, the DRI used for a collection points to a directory that contains the collection of documents as well as information used to build the collection overview and some additional metadata. DRIs can also point directly to an individual file the same way that a URL can point to either a directory or a file.

**[0086]** Since a collection typically comprises a multitude of documents, the DRI is often a directory reference rather than a reference to a particular file. For example, in an OS (operating system) such as Unix, the DRI can be a directory reference such as /usr/collection. Alternatively, the DRI can refer to a file that in turn leads to an identification of the constituent elements of a collection. In still another alternative, the DRI can be a reference to a database that stores the collection.

**[0087]** The text of the DRI 510 may comprise a string of characters that includes a random text component. This randomly (and thus, unguessable) generated text serves to prevent access to a collection because it is virtually impossible to guess.

**[0088]** The example DRI “/root/usr/collection” assumes a single-machine architecture. In a more generalized configuration of two or more machines, the DRI can include a machine name component. For example, a more accessible format such as the URL (universal resource locator) format for identifying World Wide Web (WWW) pages might be suitable. In one embodiment, the DRI constitutes the path portion of the URL. Purely by convention, the path portion uses the following naming format according to a particular embodiment of this aspect of the present invention:

.../-DDS-/ORIGIN/...,

where DDS is the name of a particular repository of collections, and ORIGIN is the fully qualified hostname of the original server for the collection identified by the DRI.

[0089] Thus, for example, a collection may be identified by the following URL:

[http://machine1.com/-msg-/machine2.com/2002/1022/298hy9y8h8#\\$30er#/1/](http://machine1.com/-msg-/machine2.com/2002/1022/298hy9y8h8#$30er#/1/)

[0090] The IP address of the machine is identified by “machine1.com.” The path portion refers to a collection stored in a repository named “-msg-.” The original copy of the collection (i.e., its place of creation) is located on a machine named “machine2.com.” Thus, in this case, “machine1” contains a copy of the collection. In one embodiment, collections are contained in directories, though other data storage conventions can be used; e.g., collections can be stored and managed in a database. The collection shown in the example above is stored in a directory called:

“/2002/1022/298hy9y8h8#\$30er#/1/.”

The pathname portion “/2002/1022” represents a date; e.g., date of creation of the collection. The string “398hy9y8h8#\$30er#” represents randomly generated text. Finally, as will be discussed below, the directory represented by the terminal pathname “/1/” refers to the first (initial, original, base, etc.) version of the collection.

[0091] In one embodiment, both the host machine (“machine1”) and the original machine (“machine2”) use the following directory structure and URL naming structure. The host machine has a directory called “-msg-” contained in its respective “root” directory for storing collections. The “-msg-” directory has a sub-directory called “machine2.com” which contains all the collections originating on “machine2.com.” Generally, a sub-directory is provided for each machine that can be an originator of a collection.

**[0092]** Given the DRI, a person or machine will have enough information to access the collection in order to add to or modify the collection.

**[0093]** Using a 2-D bar code representation of a DRI allows for automated access to the collection without requiring the user to manually enter the location. It can be appreciated of course that any machine-readable indicium can be used instead of a bar code system, including optical character recognition (OCR) of the human-readable DRI.

**[0094]** Using the MFP and/or the processing logic and the techniques described herein, it is possible to create and modify collections on a collection server. A new, empty collection can be created. A new non-empty collection can be created using available documents and media. Electronic media and paper documents can be added to existing collections. A collection can be printed. Collections can be added to or merged. Also, actions can be taken on individual media in a collection using notes or actions selected on the coversheet.

**[0095]** In one embodiment, scalable vector graphics files or SVG files are used to represent the collection overview. SVG files are a standard way of creating a visual representation on the World Wide Web and there are many viewers and tools for creating SVG. A collection preferably includes a specially name SVG file which can be used to construct an overview image for the coversheet or any display. In one embodiment, the SVG file includes information for displaying the thumbnails of individual documents and media stored in the collection.

**[0096]** Metadata about the individual files in the collection and their relationship to other files in the collection is stored preferably in an XML (extensible markup language) file. In one embodiment, this information includes image width and height,

links between images and their thumbnails and links between a document and an image representing that document. The exact format is unimportant as long as the collection server understands how to read and write the format.

**[0097]** Additional information related to the collection as a whole can also be stored in the metadata file. This information might include the time at which the message was created, the subject of the message, the name of the author of the collection, and contact information such as email addresses, fax numbers, etc. belonging to those who should be notified when a collection is altered.

**[0098]** While creating a new collection, either a printout is generated or the information about the new collection, including at least the DRI is emailed or faxed to someone. Otherwise, the DRI will be lost to all but the collection server and will not be available for adding documents because no one will have or be able to guess the DRI.

**[0099]** The MFP contacts the collection server through a network to request a new collection identifier or DRI. It should be understood that it is possible for the MFP to request identifiers in advance so that if the collection server is busy or temporarily offline, the MFP can still create new collections.

**[00100]** If the coversheet is to be printed, then the MFP composes a coversheet. In one embodiment, a header block is created including at least the date and time of the creation of the new collection. The DRI or identifier obtained from the collection server is added to the coversheet at the bottom in human-readable form and then encoded in an industry standard two-dimensional PDF417 type barcode in one embodiment and added to the upper right-hand corner of the coversheet. An SVG representing the overview is converted to image form and added to the appropriate place in the coversheet. Additional



information might also be added as deemed appropriate. The composition of the coversheet described here is one possibility but anyone skilled in the art will recognize that there are many ways to lay out or compose a coversheet that are within the scope of this invention.

**[00101]** The task of adding to an existing collection requires a collection to exist. To add to that collection at the MFP, the user uses a coversheet from the existing collection. As mentioned, each collection identifier represents a single collection but collections can change over time. In one embodiment, each time a collection changes, the last path element in the DRI is modified. Those who have access to a single collection are thereby easily given access to all versions of that collection. In one embodiment, the version name or final pathname of /0/ has a special significance and means the “latest” or “most recently created” version.

**[00102]** In one embodiment, pathname /1/ indicates the first version of the collection, /2/ represents the second version, etc. When a new collection is uploaded to the collection server, a new directory using the next integer is created. The next collection after /2/ would preferably be called /3/. In order to maintain unique version numbers, it is essential that only one device, i.e., the collection server, create the version number of final pathname. The version number cannot be created by the MFP because multiple MFPs might generate a number at the same time and choose the same name. Instead, the MFPs create a collection and upload it to a temporary directory on the collection server and when everything is uploaded, the collection server moves it into place and assigns the final pathname.

**[00103]** If the user has additional paper documents, they can be placed on an automatic document feeder. If the user has images or other documents in a memory card or some other media, the media can be placed in the appropriate reader.

**[00104]** If the user wishes to create some electronic media at the time of the creation of the new collection, the user records audio, video, still images, or other electronic media using any of the microphone, a digital camera, video camera, or other media-capturing device may be used.

**[00105]** Each DRI is associated with the page of the document or image in which it was found. However, the MFP can recognize that a page containing a DRI represents a collection. Putting a page with a DRI into any collection, new or existing, could be understood as a request to add that collection to the new collection. In other words, the page containing the DRI represents a request to add the collection pointed to by that DRI to the new collection. The overview image of that collection will be retrieved and added as a thumbnail to the new collection and the subject of that collection will be used as the title for the thumbnail.

**[00106]** Because this is a new collection, one or more new identification numbers are requested and received from the collection server. In one embodiment, only a single collection identifier is needed for a new collection.

**[00107]** Each document or page that was found to contain a DRI in machine-readable form is replaced with an image representing the collection pointed to by that DRI.

**[00108]** A thumbnail is created for each page or document or other media. The thumbnail is preferably a smaller version of the page that is similar in appearance but

smaller in storage size and in number of pixels. With recorded audio, a thumbnail is just a representation of the audio and could be a waveform or even a standard computer icon representing the audio. In the preferred embodiment, the audio could be displayed as a rectangle containing a waveform whose shape is based on the audio content and whose length corresponds to the duration of the audio recording. A video thumbnail could be a single frame or a small number of representative frames from the video composited into a single small image. Those who are skilled in the art will understand that there are many various ways of creating thumbnails to represent media. Each collection coversheet was replaced with a collection overview image that is now reduced to form a thumbnail.

**[00109]** All of the media and documents for the new collection are now added to the collection, which means that they are uploaded to the collection server and placed in the directory pointed to by the DRI of the new collection. There are many well-known protocols for uploading files to a server, including FTP, SCP, HTTP PUT. Preferably, the HTTP PUT protocol is used which allows the MFP to specify the location and contents of each media file as it is being uploaded.

**[00110]** The thumbnails representing the new media items are arranged in the collection overview. The thumbnails are placed in an appropriate manner within the overview, expanding the overview size if necessary.

**[00111]** The SVG file representing the overview is written and uploaded to the collection server and all of the thumbnails are uploaded.

**[00112]** One method for placing thumbnails is to find a place in the overview image where the thumbnail can be positioned where it will not overlap any other thumbnail. An exhaustive search – moving the thumbnail to different positions within

the overview and looking for overlaps with other thumbnails – is too slow. Another approach is to reduce the problem to that of placing a single point. This can be done by reducing the size of the overview by the width and height of the thumbnail to be placed and enlarging the existing thumbnails by the same dimensions. The origin of the new thumbnail can be placed anywhere within the remaining space in the overview without overlapping existing thumbnails. This is known as a “configuration space” approach because instead of finding a new thumbnail location in the original two dimensional space of the overview, a new “available-space” region is calculated in which the origin of the thumbnail is placed instead of the entire thumbnail. Configuration space techniques for interference checking are well known in the field of robotics and path planning.

**[00113]** The size of the thumbnail to be added to the overview is determined. Thumbnail sizes are usually measured in pixels. Often thumbnails are chosen to be some standard size – chosen so that they neither the width nor height is larger than a certain maximum size – perhaps 150 pixels for standard display resolutions or two inches for printed thumbnails. Since some images might have a very large or very small aspect ratio. It might be more appropriate to limit the thumbnail to a maximum area – square pixels or square inches – rather than a maximum width and height.

**[00114]** Scaling an image so that it contains no more than some total number of pixels instead of restricting the width and height to be less than some maximum improves the overall appearance of the thumbnails and is the preferred method of selecting a thumbnail size. However, any method for choosing thumbnail sizes can be used for the present invention.

**[00115]** In one embodiment, a single bounding box for all the thumbnails previously placed on the overview is calculated and the origin of the new thumbnail is placed outside of that bounding box. It is also possible and understood by extension that instead of calculating just a single bounding box, an individual bounding box for each thumbnail may be calculated and extended so that the new thumbnail can be placed in any available position in the overview. This is well understood by those experienced with path planning algorithms and would be analogous to allowing a machine to travel between obstacles instead of requiring the robot to go around all the obstacles.

**[00116]** Adding a second new thumbnail now to the overview could be accomplished. However, instead of adding one bounding box to cover all the thumbnails, simply adding a single box representative of the newly added thumbnail is the preferred approach. This box is calculated to be the size of the newly added thumbnail and then is extended up and to the left by the width and height of the thumbnail to be added, just like the first bounding box.

**[00117]** All new thumbnails are uploaded to the collection server as well as the new overview description file and metadata file.

**[00118]** Modifying the overview could be accomplished using an object-based drawing tool like those available in Microsoft's PowerPoint software or Adobe Illustrator or similar tools. These tools and techniques are well understood by those skilled in the art.

**[00119]** All modified information is sent to the collection server, including the metadata files, SVG overview file, and any changes in the collection.

**[00120]** The user may bring media to the MFP or creates it using media recording devices or the like connected to the MFP or to the network.

**[00121]** The advantage of having a machine-readable collection identifier on a coversheet is that the MFP or any device that can locate and decode machine-readable codes can determine which collection is represented by the coversheet. The user can indicate which collection the new media will be added to by typing in a collection identifier or DRI but this can be a difficult task because DRIs tend to be long random strings of characters. DRI's can be located and decoded from a scanned image or read using handheld barcode scanners if they are encoded in barcode format. Handheld scanners which read many different types of one and two-dimensional barcodes are available from many companies like Hewlett-Packard Company of Palo Alto, California, USA. They can also be read in text form using optical character recognition technology or decoded from a magnetic strip if properly encoded. If a coversheet of the collection is available, the coversheet should be placed on the MFP where it can be scanned, either in the automatic document feeder or directly on the glass platen. Alternatively, the barcode can be scanned using a handheld scanner. If the barcode has been captured in a digital image, perhaps using a digital camera, the camera can be directly connected to the MFP or a memory card from the camera can be plugged into a card reader. There are many other methods for presenting the MFP with a machine-readable DRI and those methods and techniques are not enumerated herein because they are understood by those skilled in the art.

**[00122]** In one embodiment, a machine-readable DRI is presented as part of the coversheet of the collection. In one embodiment, the DRI is contained in PDF417 format

two-dimensional barcode on the coversheet and the coversheet is placed on an automatic document feeder (ADF) of the MFP. Additional documents or pages to be added to the collection are placed behind the coversheet. The additional pages can be any document pages or they can be coversheets of other messages.

**[00123]** Each of the documents and media is searched for a machine-readable DRI. When a bar-coded DRI is scanned using a handheld scanner, the DRI can be stored in the memory of the MFP so that it can be accessed when it is time to determine which collection to add the new media to. If the ADF or a platen has been used to scan in a coversheet or if the DRI is contained in an image from the digital camera, the DRI will have to be read from the scanned or captured image. Either source of a DRI is acceptable and typically, if there is no DRI held in a memory due to hand scanning of a coversheet, the first scanned sheet or first image will contain the DRI. Those skilled in the art will recognize that there are many ways of providing the DRI to the MFP an exhaustive list need not be provided.

**[00124]** The entire image media including images that are scans of document pages is searched for machine-readable codes. Typically, when adding a page or document to a collection, the image of that page is added to the collection storage area and a thumbnail is added to the overview. If that page happens to contain a machine-readable DRI then based on the users preference, instead of adding the page to the collection, the collection that the DRI represents can be added to the collection. For each page or image containing a DRI, the “page add” request is converted into a “collection add” request with the appropriate DRI representing the collection.

**[00125]**        Thumbnails are created for all of the new images, documents, pages, and media. For those pages that represent other collections, thumbnails are made for the collections instead of the page. All collected media is uploaded to the collection server.

**[00126]**        It is important that the existing collection be changed in a way that makes the current state or version of the collection available at a later time. The new media should not be placed in the same storage area as the existing collection.

**[00127]**        Typically, new media and thumbnails in a collection are uploaded to a staging area on the collection server. The staging area is associated with the collection identifier but doesn't have a permanent final pathname. As soon as all of the information has been uploaded and is complete, the collection server moves the collection into a final directory or storage area with a permanent final pathname. The permanent final pathname is usually the next integer after the most recently uploaded collection.

**[00128]**        The thumbnails representing the new media are added to the collection overview.

**[00129]**        The thumbnails, metadata, and the overview SVG file, are uploaded to the staging area in the collection server. All changes and modifications are finally uploaded to the collection server and at this point, the server has everything required to move the collection out of the staging area and into the final directory upon assigning a version number.

**[00130]**        A collection server can keep a mapping of collection identifiers to collection directories.

**[00131]**        Whereas many alterations and modifications of the present invention will no doubt become apparent to a person of ordinary skill in the art after having read the



foregoing description, it is to be understood that any particular embodiment shown and described by way of illustration is in no way intended to be considered limiting.

Therefore, references to details of various embodiments are not intended to limit the scope of the claims that in themselves recite only those features regarded as essential to the invention.